

APPENDIX A

LAYOUTOBJECTS

```
5      This entity lists all of the layout objects that
      can be used in a header, footer, segment, or grid.
-->
<!ENTITY % HEADEROBJECTS
"ACCUMULATOR|HEADERFIELD|AGGREGATE|LABEL|DELIMITER|RECTANGLE|IMAGE">
<!ENTITY % FOOTEROBJECTS
10  "ACCUMULATOR|HEADERFIELD|AGGREGATE|LABEL|DELIMITER|RECTANGLE|IMAGE">
<!ENTITY % SEGMENTBODYOBJECTS
"GRID|LABEL|ACCUMULATOR|AGGREGATE|HEADERFIELD|DELIMITER|RECTANGLE|IMAGE
">
<!ENTITY % GRIDBODYOBJECTS "FIELD">
15  <!ENTITY % Left "left">
<!ENTITY % Center "center">
<!ENTITY % Right "right">
<!ENTITY % Money "money">
<!ENTITY % AlignmentValues "(%Left;|%Center;|%Right;|%Money;)">
20  <!ENTITY % AlignmentDefault "%Left;">
<!ENTITY % Int "int">
<!ENTITY % String "string">
<!ENTITY % Date "date">
<!ENTITY % Numeric "numeric">
25  <!ENTITY % Units "units">
<!ENTITY % ExpressionTypes "(%Int;|%String;|%Date;|%Numeric;|%Units;)">
<!ENTITY % ExpressionTypeDefault "%String;">
<!--
ArgumentTypes; while only INT and string are supported herein, this is
30  not intended to be limiting.
-->
<!ENTITY % ArgumentTypes "(%Int;|%String;)">
<!ENTITY % True "true">
<!ENTITY % False "false">
35  <!ENTITY % Boolean "(%True;|%False;)">
<!ENTITY % Sum "sum">
<!ENTITY % Count "count">
<!ENTITY % First "first">
<!ENTITY % Last "last">
40  <!ENTITY % Min "min">
<!ENTITY % Max "max">
<!ENTITY % Average "average">
<!ENTITY % OperationValues
"(%Sum;|%Count;|%First;|%Last;|%Min;|%Max;|%Average;)">
45  <!ENTITY % Operation '
      operation      %OperationValues;      "%Sum;"
      '>
<!ENTITY % FirstOrNewPage "firstnp">
<!ENTITY % Always "always">
50  <!-- for First we "borrow" the same constant used for operations -->
<!ENTITY % DisplayRuleValues "(%First;|%FirstOrNewPage;|%Always;)">
<!ENTITY % DisplayRule '
      displayrule    %DisplayRuleValues;      "%Always;"
      '>
55  <!ENTITY % Odd "odd">
```

```

<!ENTITY % Even "even">
<!ENTITY % Any "any">
<!ENTITY % PageStartValues "(%Odd;|%Even;|%Any;)">
<!ENTITY % PageStart '
5   pagestart      %PageStartValues;      "%Any;"
   '>
<!ENTITY % HtmlStream "HTML">
<!ENTITY % HpStream "PCL">
<!ENTITY % AfpStream "AFP">
10  <!ENTITY % ScreenStream "SCREEN">
    <!ENTITY % PdfStream "PDF">
    <!ENTITY % StreamEnum
    "(%HtmlStream;|%HpStream;|%AfpStream;|%ScreenStream;|%PdfStream;)">
    <!ENTITY % StreamAttr '
15   type      %StreamEnum; #REQUIRED
    '>
    <!ENTITY % Portrait "portrait">
    <!ENTITY % Landscape "landscape">
    <!ENTITY % OrientationEnum "(%Portrait;|%Landscape;)">
20  <!ENTITY % OrientationAttr '
    orientation %OrientationEnum; "%Portrait;"
    '>
    <!-- right now only zero or one stream (and it should be HTML) -->
    <!ENTITY % StreamElement "STREAM?">
25  <!ENTITY % Identifier '
    id          ID          #IMPLIED
    '>
    <!ENTITY % Name '
    name         CDATA         ""
30  '>
    <!ENTITY % Type '
    type      %ExpressionTypes; "%ExpressionTypeDefault;"
    '>
    <!ENTITY % Value '
35  value     CDATA ""
    '>
    <!ENTITY % ArgumentType '
    type      %ArgumentTypes; "%Int;"
    '>
40  <!--
    x          = horizontal placement of the item (from which alignment is
    derived)
    relativey  = relative vertical displacement of the item from previous
    item
45  y          = absolute vertical displacement of the item
    height     = height of the item
    width      = width of the item's "bounding box"
    alignment  = alignment of item within bounding box
    -->
50  <!ENTITY % Location '
    x          CDATA          #REQUIRED
    relativey  CDATA          "0"
    y          CDATA          "0"
    height     CDATA          "1"
55  width      CDATA          "0"
    alignment  %AlignmentValues; "%AlignmentDefault;"
    '>

```

```

<!ENTITY % LocationZeroHeight '
  x          CDATA          #REQUIRED
  relativey  CDATA          "0"
  y          CDATA          "0"
5  height    CDATA          "0"
  width      CDATA          "0"
  alignment  %AlignmentValues; "%AlignmentDefault;"
'>
<!--
10  yadjust    = yadjustment parameter for fine-tuning positioning
-->
<!ENTITY % YAdjust '
  yadjust    CDATA          "0"
'>
15  <!--
    leftmargin = left margin on page
    pagewidth  = width of the page
    rightmargin = right margin on page
    topmargin  = top margin on page
20  pageheight = height of the page
    bottommargin = bottom margin on page
-->
<!ENTITY % PageSettings '
  pagesize    CDATA          ""
25  duplex     CDATA          "0"
  leftmargin  CDATA          "0"
  pagewidth   CDATA          #REQUIRED
  rightmargin CDATA          "0"
  topmargin   CDATA          "0"
30  pageheight CDATA          #REQUIRED
  bottommargin CDATA          "0"
'>
<!--
  visible     = default visibility of the item
35  -->
<!ENTITY % Visibility '
  visible     %Boolean;      "%True;"
'>
<!ENTITY % Floating '
40  floating   %Boolean;      "%False;"
'>
<!--
  sortorder = order of this item relative to other items for sorting
records
45  sortascending = true if records will be sorted in ascending order on
this item
-->
<!ENTITY % Sorts '
  sortorder    CDATA          "0"
50  sortascending %Boolean; "%True;"
'>
<!ENTITY % Repeat '
  repeat       %Boolean;      "%True;"
'>
55  <!ENTITY % FormatString '
  format       CDATA          ""
'>

```

```

<!ENTITY % ValueFormatString '
    valformat CDATA ""
'>
5 <!ENTITY % RequiredFormatString '
    format CDATA #REQUIRED
'>
<!ENTITY % Font '
    font CDATA "1"
'>
10 <!ENTITY % Line '
    line CDATA "1"
'>
<!ENTITY % Extents '
    xextent CDATA "0"
15 yextent CDATA "0"
'>
<!ENTITY % Shading '
    shading CDATA "0"
'>
20 <!ENTITY % NewPage '
    newpage %Boolean; "%False;"
'>
<!ENTITY % TextWrap '
    wrap %Boolean; "%False;"
25 '>
<!ENTITY % Exclude '
    exclude %Boolean; "%True;"
'>
30 <!ENTITY % Winpath '
    winpath CDATA ""
'>
<!ENTITY % Filename '
    filename CDATA ""
'>
35 <!-- An example selection of encodings -->
<!ENTITY % EncodeUTF8 "UTF-8">
<!ENTITY % EncodeUTF16 "UTF-16">
<!ENTITY % EncodeISO88591 "ISO-8859-1">
40 <!ENTITY % EncodeEnum "(%EncodeUTF8;|EncodeUTF16;|EncodeISO88591;)">
<!ENTITY % EncodeAttr '
    charset %EncodeEnum; #REQUIRED
'>
<!--
45 LAYOUT

    Root-level element object.
    A layout is composed of segments. It optionally contains a
    description block.

50 Attributes:
    dtdversion Version of the DTD on which this XML file is based
    version Version # of this layout specification
-->
55 <!ELEMENT LAYOUT (DESCRIPTION?, METADATA?, NAME?, SEGMENT+)>
<!ATTLIST LAYOUT
    dtdversion CDATA #FIXED "19990827"
    %Name;

```

```

        version CDATA ""
    >
    <!--
    DESCRIPTION
5      A description is a block that contains information about a layout
    -->
    <!ELEMENT DESCRIPTION (TITLE?, AUTHOR?, MANAGER?, COMPANY?, COPYRIGHT?,
    COMMENTS?, PROPERTY*)>
10    <!ATTLIST DESCRIPTION
        %Identifier;
    >
    <!ELEMENT TITLE (#PCDATA)>
    <!ATTLIST TITLE
15        %Identifier;
    >
    <!ELEMENT AUTHOR (#PCDATA)>
    <!ATTLIST AUTHOR
        %Identifier;
20    >
    <!ELEMENT MANAGER (#PCDATA)>
    <!ATTLIST MANAGER
        %Identifier;
    >
25    <!ELEMENT COMPANY (#PCDATA)>
    <!ATTLIST COMPANY
        %Identifier;
    >
    <!ELEMENT COPYRIGHT (#PCDATA)>
30    <!ATTLIST COPYRIGHT
        %Identifier;
    >
    <!ELEMENT COMMENTS (#PCDATA)>
    <!ATTLIST COMMENTS
35        %Identifier;
    >
    <!ELEMENT PROPERTY (#PCDATA)>
    <!ATTLIST PROPERTY
        %Identifier;
40        %Name;
    >
    <!--
        METADATA
45      Contains information about the generated invoice at the invoice
        level rather than the field or segment level.
    -->
    <!ELEMENT METADATA (DTDPATH?, ENCODING?)>
    <!ELEMENT DTDPATH (#PCDATA)>
50    <!ELEMENT ENCODING (#PCDATA)>
    <!ELEMENT NAME (#PCDATA)>
    <!--
        SEGMENT
55      A section as an optional header and footer, which appear on each
        page of the invoice, an optional visibility filter, and any number of
        layout objects.

```

```

-->
<!ELEMENT SEGMENT (HEADER?, (%SEGMENTBODYOBJECTS;)*, FOOTER?,
VISIBILITY?, %StreamElement;)>
<!ATTLIST SEGMENT
5      %Identifier;
      %Name;
      %PageSettings;
      %OrientationAttr;
      %PageStart;
10     %Visibility;
>
<!--
    HEADER

15     Header object for segments and grids.
    A header is a collection of layout objects.
-->
<!ELEMENT HEADER (%HEADEROBJECTS;)*>
<!ATTLIST HEADER
20     %Identifier;
      %Visibility;
>
<!--
    FOOTER

25     Footer object for segments.
    A footer is a collection of layout objects.
-->
<!ELEMENT FOOTER (%FOOTEROBJECTS;)*>
<!ATTLIST FOOTER
30     %Identifier;
      %Visibility;
      %Floating;
>
35 <!--
    DELIMITER

    A delimiter is a line drawn between two points

40 -->
<!ELEMENT DELIMITER (VISIBILITY?)>
<!ATTLIST DELIMITER
      %Identifier;
      %Name;
45     %LocationZeroHeight;
      %YAdjust;
      %Repeat;
      %Visibility;
      %Line;
50     %Extents;
>
<!--
    RECTANGLE

55     A rectangle is a line drawn between two points
-->

```

```

<!ELEMENT RECTANGLE (VISIBILITY?)>
<!--
  IDENTIFIER
  NAME
  LOCATIONZEROHEIGHT
  YADJUST
  REPEAT
  VISIBILITY
  LINE
  SHADING
  EXTENTS
-->
<!--
  IMAGE
  A IMAGE is an object which will be rendered according
  to a graphics file.
-->
<!ELEMENT IMAGE (VISIBILITY?, %StreamElement;)>
<!--
  IDENTIFIER
  NAME
  WINPATH
  FILENAME
  EXCLUDE
  LOCATION
  REPEAT
  VISIBILITY
-->
<!--
  LABEL
  A label is a fixed value of a specific type.
-->
<!ELEMENT LABEL (VISIBILITY?, %StreamElement;)>
<!--
  IDENTIFIER
  NAME
  TYPE
  VALUE
  LOCATION
  FORMATSTRING
  REPEAT
  VALUEFORMATSTRING
  VISIBILITY
  FONT
  TEXTWRAP
-->
<!--
  FIELD
  A field is a value of a specific type derived from an expression,
  with an optional applied visibility filter.
-->
<!ELEMENT FIELD (EXPRESSION+, VISIBILITY?, %StreamElement;)>
<!--
  IDENTIFIER

```

```

        %Name;
        %Location;
        %Font;
        %FormatString;
5      %Sorts;
        %Visibility;
        %DisplayRule;
        %TextWrap;
    >
10  <!--
    HEADERFIELD

    A header field is a string value derived from a format string and
    zero or more expressions, with an optional applied visibility filter.
15  -->
    <!ELEMENT HEADERFIELD (EXPRESSION*, VISIBILITY?, %StreamElement;)>
    <!ATTLIST HEADERFIELD
        %Identifier;
        %Name;
20      %Location;
        %Repeat;
        %RequiredFormatString;
        %Font;
        %Visibility;
25      %TextWrap;
    >
    <!--
    ACCUMULATOR

30  An accumulator is a summation of an expression, with an optional
    applied visibility filter.
    An accumulator acts like a regular field, in that a single expression
    can generate
    multiple values, but for accumulators, only the final accumulation is
35  displayed
    on the invoice; i.e., the individual expression values themselves are
    not displayed.
    -->
    <!ELEMENT ACCUMULATOR (FILTER?, EXPRESSION, VISIBILITY?,
40  %StreamElement;)>
    <!ATTLIST ACCUMULATOR
        %Identifier;
        %Name;
        %Location;
45      %Operation;
        %Font;
        %FormatString;
        %Repeat;
        %Visibility;
50      %TextWrap;
        currency_format CDATA ""
    >
    <!--
    AGGREGATE
55

    An aggregate is an aggregation of multiple instances of a single
    field or multiple fields, such

```


as the rows of a column in a grid. The aggregate specifies an operation to perform on the instances of the field, such as "sum" or "count."

-->

5 <!ELEMENT AGGREGATE (FIELDREF+, VISIBILITY?, %StreamElement;)>
<!ATTLIST AGGREGATE

%Identifier;
%Name;
%Location;
10 %Font;
%FormatString;
%Repeat;
%Visibility;
%Operation;
15 %TextWrap;
currency_format CDATA ""

>

<!--

FIELDREF

20

A field reference is the name of a field within this layout specification.

-->

<!ELEMENT FIELDREF (#PCDATA)>

25

<!ATTLIST FIELDREF
%Identifier;

>

<!--

GRID

30

A grid is a "table." It has a header and footer, and contains a grouping, one or more nested grids, or layout objects. A visibility filter can be applied to the grid.

-->

<!ELEMENT GRID (FILTER?, HEADER?, (GROUPING | ((%GRIDBODYOBJECTS;)* | GRID+), FOOTER?, VISIBILITY?)>

<!ATTLIST GRID

40

%Identifier;
%Name;
%Location;
%Visibility;
%NewPage;

45

>

<!--

GROUPING.

50

A GROUPING is a partition of a grid's rows. It has a header and footer, and contains either another GROUPING, or, at the most nested level, layout objects.

Rows of a grid are placed into groups according to the GROUPING-FILTER

55

for each GROUPING. The groups are then sorted according to the EXPRESSION.

-->

```

<!ELEMENT GROUPING (GROUPING-FILTER, EXPRESSION?, HEADER?, (GROUPING |
  ((%GRIDBODYOBJECTS;)* ) | GRID+), FOOTER?)>
<!ATTLIST GROUPING
  %Identifier;
5    %Name;
    %NewPage;
>
<!--
10  EXPRESSION

    An expression is an evaluated function. Here, it is the name
    of the function to evaluate.

    Attributes:
15    format      = format identifier for the expression
-->
<!ELEMENT EXPRESSION (#PCDATA)>
<!ATTLIST EXPRESSION
20    %Identifier;
    %Sorts;
    format CDATA ""
>
<!--
25  VISIBILITY

    Visibility is a kind of filter. The data of VISIBILITY is the name
    of the function to call to determine whether an item should be
    displayed on the invoice or not.
-->
30 <!ELEMENT VISIBILITY (#PCDATA | FILTERARGUMENT)*>
<!ATTLIST VISIBILITY
    %Identifier;
>
<!--
35  STREAM

    Printer driver-specific stream for layout objects. The stream text
    is used as a format string for the layout object, i.e., inside the
    printer driver, if it supports streams, the value of an evaluated
40 layout object is then plugged into the stream text and then sent
    to the output.
-->
<!ELEMENT STREAM (#PCDATA)>
<!ATTLIST STREAM
45    %StreamAttr;
>
<!--
    FILTER

50  A filter is an evaluated function. It specifies what rows will be
    used to create a view for a grid.
-->
<!ELEMENT FILTER (#PCDATA | FILTERARGUMENT)*>
<!ATTLIST FILTER
55    %Identifier;
>
<!ELEMENT FILTERARGUMENT (#PCDATA)>

```

```

<!-- ATTLIST FILTERARGUMENT
      %ArgumentType;
      %Value;
-->
5  <!--
      GROUPING-FILTER

      A grouping filter is another kind of filter. It determines an
exhaustive
10  partition of the grid's view into subviews.
      -->
      <!-- ELEMENT GROUPING-FILTER (#PCDATA) -->
      <!-- ATTLIST GROUPING-FILTER
            %Identifier;
15  -->

```